

LOOK AT ALL THESE!

# NEW OUTPUT

from

PRENTICE-HALL



STATEMENT	
1	EXOGENOUS EVENT ORDERIN
2	SAVE EVENT CARD
3	CREATE CRDR
4	LET DATE (ORDERS) = TIME
5	READ N
6	TYPE N (14)
7	DO TO 10, FOR I = (1) (1)
8	CREATE DESTIN
9	READ FILE (DESTIN), PRIME (DESTIN)
10	FORMAT (14, H3.2)
	FILE DESTIN IN ROUT(ORDER)
	LOGIC (14)
	CALL ASRBL(ORDER)
	RETURN
	END

Fig. 5 - Exogenous Event Routine Describing the Receipt of an Order

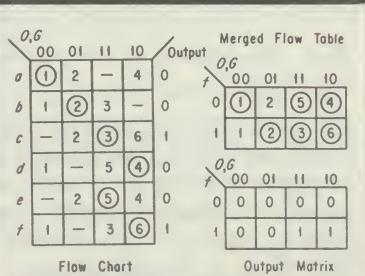


```

procedure Forco(a, b, n, fct, cs) result : (ab);
value a, b, n;
real a, b;
integer n;
array ab;
real procedure fct, cs;

begin integer i, k; real p, q, r, s;
p := (b - a)/(2 * n);
q := 6.283185307/(b - a);
for k := 0 step 1 until n - 1 do
begin
  r := k * q; s := 0;
  for i := 0 step 1 until 2 * n - 1 do
    s := s + fct(a + i * p) * cs(r * (a + i * p));
  ab[k] := s/n;
end
end

```



Four experts show you programming techniques  
for any existing ALGOL compiler in . . .

## AN INTRODUCTION TO ALGOL

By Dr. F. L. BAUER, Dr. RICHARD BAUMANN, both of the Munich Institute of Technology, Germany, MANUEL FELICIANO, Mathematician, Oak Ridge National Laboratory and Dr. K. SAMELSON, Institute of Advanced Mathematics, Johannes Gutenberg University, Mainz, Germany.

### LEARN TO PROGRAM quickly and easily

During the past five years, nearly 500 European and American students have already learned ALGOL from this one self-contained book. Now, you too, can quickly master ALGOL and write complete programs sooner than you expected. And, this self-study guide makes it easy for you, as it emphasizes the practical use of the language and arranges the topics in an order which enables you to easily recognize the systematic construction of ALGOL.

### GRASP CONCEPTS AND TECHNIQUES with little effort

Step-by-step, this self-contained manual guides you through a series of carefully thought-out examples to help you grasp the basic concepts while learning good programming techniques. Each illustrative example leads into the next so you progress logically, at your own speed. This book was especially written to teach you, the prospective programmer, the ALGOL you will need to use the computer as a tool.

Nothing is overlooked or omitted: ALGOL thoroughly treats blocks and procedures; explains jumps and loops; covers propositions and conditions and designational expression — You'll be amazed at how quickly and easily you learn. This book even helps you eliminate errors before they are made by including valuable exercises which point out common programming mistakes.

### EXPERIENCE OF AUTHORS an outstanding feature

This book is based on five years of extensive practical experience in compiling and teaching ALGOL both in Europe and in America. Some of the installations the authors were working on have been operated exclusively by means of ALGOL since 1959 when they had their translators running. The book is a revised and improved translation of the ALGOL Manual distributed by the ALCOR Group. (ALCOR is a cooperative group of ALGOL compiler builders and users with which the authors are associated. It has accomplished the task of having translators in Europe ready for almost any computer used there for technical and scientific calculations, and it has promoted standardization of ALGOL with respect to common hardware coding.)

Dr. Bauer is a co-initiator of ALGOL and the Director of the Mathematical Institute and the Computing Center of the Munich Institute of Technology. Dr. Baumann is the author of the ALGOL Manual of the ALCOR Group and has instructed many training courses on ALGOL, arranged by several institutes and computer manufacturers since 1958. Manuel Feliciano has been intimately connected with ALGOL since its inception (1957). He was one of the builders of a

translator for the ORACLE computer at Oak Ridge National Lab., where he has taught ALGOL. He has also been involved in ALCOR group discussions. Dr. Samelson has taught ALGOL at the Johannes Gutenberg University.

### Cross-section of CONTENTS for time-saving ALGOL Manual

INTRODUCTION. Mode of operation and capability of digital computers. Programming. Coding. Introductory examples. I. ALGORITHMIC LANGUAGE ALGOL — ELEMENTARY PART. Basic symbols of the language. Arithmetic expressions. Construction of the program. Loops. The conditional statement. Jumps. II. ALGORITHMIC LANGUAGE ALGOL — FURTHER CONSTRUCTIONS. Block structure. Propositions and conditions. Designational expressions. Procedures. Procedures calling themselves. Examples. APPENDIX consists of the Revised Report of the Algorithmic Language ALGOL 60.

176 pp. 6 x 9" illus. (Dec. 1963) \$8.

Thoroughly master machine principles while teaching yourself flow charting and programming with . . .

## PROGRAMMING AND WIRING THE UNIVAC 1004

By JAMES A. SAXON, Saxon Research Corp., San Diego, Calif. and RICHARD W. SENSEMAN, San Fernando Valley State College, Northridge, Calif.

This self-instructional manual has been study-tested in UNIVAC classes to teach beginners basic programs, advanced routine techniques, and the use of the standard report generator. With this one self-contained book, you too can start from scratch teaching yourself flow-charting, programming and wiring on the UNIVAC 1004. Quickly and easily, you'll master machine principles while learning to use charts and forms expertly.

You need no special training or background — you need no previous knowledge of data processing or computer programming. In fact, the less you know about the field the better, for then you won't have to break established habits or unlearn any preconceived ideas.

With this manual you learn by doing! You actively participate in the learning process by actually working out over 200 carefully selected problems. You always get immediate feed-back of the correct answers to check your own progress and insure retention of the material. The problems and answers are further reinforced with step-by-step explanations, examples, wiring diagrams, and other illustrative at-a-glance aids. In addition, you can double check your understanding of previously learned material by scoring yourself on frequent short quiz sessions.

You'll find each lesson to be brief, yet not so brief as to bore you. You'll enjoy studying and learning as you can progress at your own pace and at your own convenience. Get a copy in your hands and master the UNIVAC 1004 by yourself, more quickly and easily than you ever believed possible.

approx. 200 pp. 200 illus. 8½ x 11" (Dec. 1963) \$6.  
over please . . .

Another sure BEST-SELLER by IVAN FLORES for everyday professional application

Here is THE ever-ready, detailed description of all the known high-speed methods, algorithms and shortcuts by which the computer does arithmetic

all here in IVAN FLORES' brand new —

# LOGIC of COMPUTER ARITHMETIC

(a for-all-times coverage of high-speed parallel binary computer arithmetic)

## New treatments to keep you up-to-date

This totally new, advanced book deals exclusively with logic, not circuitry, and provides many first-time descriptions (in depth) of areas you yourself may cover in daily work. For example: floating point arithmetic is, for the first time, completely dissected; thorough treatment is given to special purpose arithmetic units; as well as the first presentation of its kind of residue arithmetic and its applications.

## More descriptions of high-speed methods and shortcuts to increase your working knowledge

Other special treatments completely cover these areas: binary number representations; basic high-speed adders; short-cut multiply and divide algorithms; and square root and polynomial logic.

IVAN FLORES is a private consultant and Associate Professor at the Polytechnic Institute of Brooklyn. He is the author of best-selling COMPUTER LOGIC described at the top of the next column on this page.

## Here is just a partial list of CONTENTS to give you an idea of the in-depth coverage of the new LOGIC OF COMPUTER ARITHMETIC

1. INTRODUCTION. Computer system structure. Processing & arithmetic. Symbology. 2. BINARY REPRESENTATIONS & ARITHMETIC. Addition 1's complement. Subtraction. 3. MULTIPLICATION & DIVISION IN DIFFERENT REPRESENTATIONS. 1's complement multiplication, division. 2's complement multiplication, division. 4. SIMPLE FAST ADDERS. The synchronous adder. The self-timing adder. 5. THE CARRY LOOKAHEAD ADDER. The sum block. Further Levels. 6. THE CONDITIONAL SUM ADDER. First level. Generate Case. 7. THE SIMPLE ARITHMETIC UNIT FOR PARALLEL BINARY 2's COMPLEMENT NOTATION. Compound addition & subtraction. Multiplication. Division. Timing Considerations. 8. MULTIPLICATION. Carry-save adders. Multiplier multiples, pairs, triplets, quadruplets & more. 9. TIMING. MULTIPLICATION. Multiplier pairs, positive numbers. Pairs, Scanning Multiplier Left to Right. Triplets, hardware. 10. MULTIPLY, SHIFTING OVER 0's & 1's. Logic equations, improved method, positive multipliers, negative multipliers. Full logic equations, improved method, signed multipliers. 11. COMPOUND MULTIPLIERS; COMPARATIVE SUMMARY. Considerations, timing specifications. 12. DIVISION I. Bit-at-a-time division. Shifting over 0's. Quotient lookahead principles, positive remainder, principles, hardware. Lookahead, multiple adders. 13. DIVISION II. Shifting over 0's & 1's, signed numbers. Termination, shifting over 0's & 1's. Quotient development, combined look-ahead, 0's & 1's shift. 14. DIVISION III. Example 3/4, 1, 3/2 divisor multiples. Quotient bit development 3/4, 1, 3/2 divisor multiples. Hardware 3/4, 1, 3/2 divisor multiples. 15. FLOATING POINT NUMBER ADDITION & SUBTRACTION. Register configuration & task analysis, addition & subtraction. 16. FLOATING POINT MULTIPLICATION & DIVISION. Problem exponent, multiplication. Auxiliary multiplication manipulation. 17. SPECIAL ARITHMETIC & COMPOUND ARITHMETIC. The decimal root. The non-restoring binary root. The series evaluator, philosophy, hardware. 18. RADIX ARITHMETIC. Radix or modulus notation. Modulo adders. Overflow. The future. BIBLIOGRAPHY.

493 pp.

126 illus.

6 x 9"

(1963)

\$15.

## COMPUTER LOGIC:

### The Functional Design of Digital Computers

Now you can learn all about computers — just how they work and how they are put together by studying the complete block diagram analysis of each operational unit of a digital computer given in this book. You will learn easily and quickly aided by explicit descriptions and illustrations. By the time you finish this book you will understand how a control unit or an input/output unit functions; how logical elements are combined to make each unit do its job. A complete illustrative example embodies the main ideas of the book pointing out to you the varied phases of computer work.

458 pp. illus.

6 x 9"

(1960)

\$14.

No previous knowledge is required for the design of the switching logic in a modern digital computer — or any digital computer, when you use

## THE LOGIC DESIGN OF TRANSISTOR DIGITAL COMPUTERS

By GERALD A. MALEY, Development Engineer & Manager of Exploratory Logic Group, and JOHN EARLE, Staff Engineer & Manager of Exploratory Logic-Systems, IBM.

The newest usable concepts in circuit design plus the latest logic design methods are uniquely combined with all the practical knowledge and many design hints that will be useful in your own work.

**Original contributions show you how to implement NOR logic** — With the Stroke function in its several NOR circuit forms as the most popular transistor switch in modern computer design, this is the only book treating this function as pivotal by developing a Stroke Algebra in place of the more elementary Boolean Algebra.

**Gives you practical discussions of arithmetic design** — A modern discussion of the arithmetic systems, including carry look-ahead techniques, frames the developed mathematical techniques with the practical aspects of the most important combinational problems in a computer.

**Helps you to get a quick, new insight into logical foundations** — Demonstrated with many worked-out examples, the introduction of minimization techniques such as the Karnaugh Map, as equivalent interpretations of the axioms of Boolean Algebra is a new approach which develops an early understanding of the logical foundations.

**More benefits include:** the first treatment of asynchronous sequential circuit theory as developed in terms of electronic circuits; and a permanent handbook-reference to the most common sequential circuit problems you frequently encounter in the design of an actual computer as worked examples, with optimal solutions.

**There's a wealth of professional know-how in these Chapters:** 1. Transistors & Circuits. 2. Number Systems & Codes. 3. Boolean Algebra. 4. Logic Design. 5. Minimization Methods. 6. Implementing NOR & NAND Logic. 7. Arithmetic Operations. 8. Asynchronous Sequential Circuits I. 9. Asynchronous Sequential Circuits II. 10. Asynchronous Sequential Circuits III. 11. Self-Timing Asynchronous Circuits.

322 pp.

340 illus.

6 x 9"

(1963)

\$14.

**Ideal for either on-the-job training or as refresher references . . .**

## **COBOL: A Self-Instructional Programmed Manual**

COBOL (*Common Business Oriented Language*) is fast winning acceptance as the universal programming language for commercial application. Now, you can teach yourself all the fundamentals of COBOL in the shortest possible time. This self-contained manual enables you to write programs for a great variety of computers, not just one specific machine. A COBOL program (with small modifications) is equally applicable to any one of a variety of IBM, BURROUGHS, REMINGTON RAND, HONEYWELL, and other computers. No previous knowledge is required for you to progress, step-by-step, through each lesson. 200 clear and intelligible problems with frequent quiz sessions are posed after principles are introduced and fully explained. These practical problems are followed immediately with a feed-back of the correct answers to reinforce the learning process and to permit you to begin programming with COBOL at once.

## **SIMSCRIPT: A Simulation Programming Language**

EASY-TO-APPLY, SIMSCRIPT programs consist of a number of different kinds of both written and generated programs that enable anyone already familiar with computer programming to apply SIMSCRIPT to his work. Simulation (*and non-simulation*) with computers in a variety of fields can be accomplished quickly and efficiently as all necessary information including detailed instructions and forms are given. SIMSCRIPT reduces programming time severalfold as compared to simulations written in FORTRAN, and permits relatively easy program modification and expansion. A number of sub-programs required for simulation are generated by SIMSCRIPT either automatically or with a minimum of user specification. This new computer language provides a wide range of commands applicable to a variety of combinations and uses to reduce working time, ease modeling, allow modifications and provide flexibility for management and systems problems.

## **PROGRAMMING the IBM 1401:**

### **A Self-Instructional Programmed Manual**

NOW AT YOUR INSTANT COMMAND! • You need no previous knowledge or experience • You proceed at your own pace with small, easy-to-grasp bits, all organized in step-by-step, logical sequence • You measure your own progress and understanding by self-scoring yourself on over 200 typical computer problems • You always get immediate feed-back of correct answers to the problems plus frequent short quiz sessions — no questions are ever left unanswered • You learn to program quickly and thoroughly aided by extensive, crystal-clear explanations and examples • You avoid costly errors and time-wasting pitfalls with proven techniques and short-cuts that usually come only after long programming experience.

## **PROGRAMMING the IBM 1620**

### **Use With Either Card or Tape 1620**

You will be able to write, assemble and run SPS programs on the 1620 by the time you're one-third of the way through this new book. With no previous knowledge you can master the 1620 with this self-contained manual which not only gives you all the major 1620 programming methods, but also the Symbolic Programming System (SPS), FORTRAN, GOTRAN, and the Interpretative Programming System (IPS). Nine appendices provide you with a comprehensive handbook of 1620 operations, character codes, instructional timings, and a summary of the one-pass and two-pass SPS languages plus operation of the corresponding processors. Can be used with either card or tape 1620.

## **PROGRAMMING the IBM 7090:**

### **A Self-Instructional Programmed Manual**

This book alone gives you all the essentials for actually programming the IBM 7090 (also applicable to the IBM 709/7094 computers, and to some degree to the IBM 704). Organized along teaching machine principles, using simple language and easy-to-follow explanations, the manual contains over 140 answered problems, extensive examples, self-scoring quizzes. Among the vital specifics you'll master are: the functions of the data channel; the characteristics of card and tape devices; the three phases of computer processing: input, computation & output; and the four steps of planning: analysis of the application, planning and sequencing steps to be used, writing the instructions & determining which areas of storage will be used for various purposes.

By **JAMES A. SAXON**, Saxon Research Corp., San Diego, California

**EXAMINE SOME OF THESE SELF-TEACHING COBOL LESSONS:** General Information. Program Divisions. Format for Program Writing, Printing. Flow Charting Symbols. Division Formats, COBOL Words. Constants, Literals, Collating Sequence. Elements of a Sentence, Expressions. Implied Subjects & Operators, Statements. Compiler Directing Instructions. Input-Output Instructions. Read & Write Statements. Data Movement Instructions. Relating Flow Charts to COBOL. General Statement. Levels. Record Description Entries. Working Storage Section. Listing of COBOL Words.

200 pp. illus. 8½ x 11" (1963) paperbound \$6.

By **HARRY M. MARKOWITZ, HERBERT W. KARR**, both of California Analysis Center, Inc., and **BERNARD HAUSNER**, The Rand Corp.

**CROSS-SECTION OF CONTENTS FOR TIME-SAVING SIMSCRIPT MANUAL:** Basic Concepts. Variables & Identification Numbers. An Example of a Simscript Source Program. Entity Operations. Arithmetic & Control Commands. Decision Commands. Input & Output Commands. Miscellaneous Commands. Event Routines & Other Sub-Programs. Report Generator. Automatically Generated System Variables & Functions. Compilation & Execution. Definition Cards. Initial Conditions Deck. Memory Layout.

138 pp. 8½ x 11" (1963) paperbound \$5.

By **JAMES A. SAXON**, Saxon Research Corp., San Diego, California, and **WILLIAM S. PLETTE**, United Research Services, Inc., Sierra Vista, Arizona

**A SAMPLING OF THE FULL-COVERAGE LESSONS** Computer Storage. Computer Word. Data Words. Instruction Word & Use. Reserved Areas of Storage. Input-Output Instructions. Erasing Information. Magnetic Core Storage. Data Transmission. Addition & Subtraction. Symbolic Coding Sheet. Constants & Work Areas. Character Adjusting. Variations of Read, Print & Punch. Printer Controls. Multiplication & Division. Address Modification. Writing on Magnetic Tape. Reading from Magnetic Tape. Special Features of Tape Processing. Read & Write Subroutines. System Analysis, System Programming.

224 pp. illus. 6 x 9" (1962) \$9.

By **CLARENCE B. GERMAIN**, College of St. Thomas, St. Paul, Minnesota

**CHECK THESE CHAPTER HEADINGS & APPENDICES:** Computing Fundamentals. 1620 Instructions. Operation of the 1620. The Symbolic Programming System. Address Arithmetic. Subroutines & Floating Point. Program Planning & Debugging. FORTRAN. GOTRAN. Intercom 1620. Special Features. Answers to Selected Problems. APPENDICES: Significance of P & Q Addresses. System Timings. Character Coding. Error Restart Procedures. Summary of Two-Pass SPS. Summary of One-Pass SPS. FORTRAN Summary. GOTRAN Summary. Intercom Summary.

208 pp. illus. 8½ x 11" (1962) paperbound \$6.

By **JAMES A. SAXON**, Saxon Research Corp., San Diego, California

**THESE ARE SOME OF THE LOGICALLY ASSEMBLED LESSONS:** Decimal, Octal and Binary Numbering Systems. Binary Arithmetic. Converting Octal to Decimal. Machine Words. Fixed Point Numbers. Format for Writing a Program. Floating Point Numbers. Symbolic Language, Use of Asterisk, Plus or Minus. Use of Constants & Literals. Use of Index Registers. Tape — Definitions. Input-Output Instructions & Commands. Use of Subroutines. Logical Operations (AND/OR). Sense Indicator Operations. Trapping. Sorting. Sample of a Complete Program.

210 pp. illus. 6 x 9" (1963) \$9.

Surpasses competition in comprehensive coverage  
with many valuable "assists" for you in . . .

## DIGITAL COMPUTER ENGINEERING

By Harry J. Gray, Moore School of Electrical  
Engineering, University of Pennsylvania

**LOGICAL ORGANIZATION** patterned after actual work flow  
All the topics you will face in the development of a digital computing system are presented here in the same sequence that you will encounter in daily work. GRAY gives an overall and integrated picture of the problems involved in the engineering of an electronic digital computing system.

**COMPLETE COVERAGE** to answer every question

In terms of specifics, DIGITAL COMPUTER ENGINEERING is far and away ahead of competition in its invaluable coverage and emphasis on the techniques used for the solution of digital computer engineering problems in statistical design, optimization, reliability, cross-talk, signal transmission, power transmission, and design automation.

**PROFESSIONAL "ASSISTS"** for your daily work

You will especially appreciate GRAY's contributions in the following areas which are sure to be of invaluable professional help to you:

1. The integration of procedures
2. Determination of requirements for digital computer circuits
3. Selection of design techniques
4. Cross-talk control with formulas for calculation
5. Reliability considerations
6. Material on design automation

Here is the complete picture as outlined in the  
**CHAPTER HEADINGS:**

1. The Eniac & Edvac. 2. Organizational Principles of Computer Design. 3. Combinational Logic. 4. Sequential Logic & Timing. 5. Digital Computer Circuits. 6. Digital Computer Circuit Analysis. 7. Reliability & Components. 8. Circuit Design & Optimization. 9. Pulse & DC Power Transmission Systems. 10. Applications of Computers to the Design of Computers. APPENDICES. 1. Introduction to Semiconductor Diode Theory. 2. Statistical Considerations. 3. Lossless & Almost Lossless Uniform Transmission Lines.

464 pp.      375 illus.      6 x 9"      (1963)      \$16.

## INFORMATION PROCESSING LANGUAGE-V MANUAL

By ALLEN NEWELL, Graduate School of Industrial Admin., Carnegie Institute of Technology, ET AL.

This popular book introduces you to the IPL-V, a programming language for non-arithmetic information processing. It also provides a complete reference manual for users of IPL-V.

This valuable manual provides you with a complete description of the language and step-by-step explanations of its use. The basic programming technique which made this language possible is the idea of a list and a list structure — a technique whereby the constraints inherent in the sequential ordering of memory are avoided. Although it focuses on list processing, IPL-V is a complete language containing both fixed and floating point arithmetic, input-output facilities, and adequate monitoring and debugging techniques.

244 pp.      8½ x 11"      (1961)      Paperbound \$6.

## COMPUTER PROGRAMMING HANDBOOK: Guide for Beginners

By ROBERT NATHAN, Research Group Supervisor, Space Sciences Div., Jet Propulsion Labs., Calif. Inst. of Tech. and ELIZABETH HANES, Computer and Microfilm Systems Consultant.

In a style easily read and understood, this helpful handbook leads you step-by-step through fundamental computer programming, clearly spelling out the principles involved. From no beginning knowledge, you can learn how to operate the leading computers in the field.

FOUR different computer models, representing both the binary and decimal types, are described and explained to give you a broad familiarity with computers in general. If you have to work with a machine not described, you simply follow the principles you have learned. This handbook isn't concentrated on a few, special-type problems . . . it builds a fundamental working knowledge applicable to almost every situation.

This guide casts aside the idea that you must tackle complex algebraic problems while learning the framework and use of a computer. It employs only basic arithmetic. It provides examples and problems and, when appropriate, gives a summary list of pertinent commands used for each computer. Through an understanding of the four described models and by following the emphasized principles, you can write an actual computer program.

214 pp.      illus.      6 x 9"      (1961)      \$10.

— Fill in the enclosed order card or clip and mail this coupon today! —

**Prentice-Hall, Inc., Englewood Cliffs, N. J.**

Send me the books checked below. At the end of ten days I'll either send the indicated price, plus a few cents for shipping and insurance, or return the books and owe nothing.

An Introduction to ALGOL,  
by Bauer, Baumann, Feliciano & Samelson . . . . . \$ 8.

Programming and Wiring the UNIVAC 1004,  
by Saxon & Senseman . . . . . \$ 6.  
 Logic of Computer Arithmetic, by Flores . . . . . \$15.  
 Computer Logic, by Flores . . . . . \$14.  
 The Logic Design of Transistor Digital Computers,  
by Maley & Earle . . . . . \$14.  
 COBOL: A Self-Instructional Programmed Manual,  
by Saxon . . . . . (paperbound) \$ 6.  
 SIMSCRIPT: A Simulation Programming Language,  
by Markowitz, Karr & Hausner . . . . . (paperbound) \$ 5.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ Zone \_\_\_\_\_ State \_\_\_\_\_

1

D48

**15% DISCOUNT to individuals** — When you order five or more books of the same or different titles (even jointly with associates on a single order), Prentice-Hall will bill the books at a 15% discount.

Programming the IBM 1401: A Self-Instructional  
Programmed Manual, by Saxon & Plette . . . . . \$ 9.  
 Programming the IBM 1620, by Germain . . . (paperbound) \$ 6.  
 Programming the IBM 7090: A Self-Instructional  
Programmed Manual, by Saxon . . . . . \$ 9.  
 Digital Computer Engineering, by Gray . . . . . \$16.  
 Information Processing Language-V Manual,  
by Newell, et al . . . . . (paperbound) \$ 6.  
 Computer Programming Handbook: Guide for Beginners,  
by Nathan & Hanes . . . . . \$10.

**SAVE!** Send payment with order and we'll prepay  
shipping and insurance. Same return privilege and  
refund.